Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Fall 2018

The Brew Crew
DOOM Bot

Eddie Molina
Peter Dang
Robert Villazana
Van Ta
Marco Paramo
Matthew Draft

Revision History

Revision	Date	Author(s)	Description
0.1	9.21.2018	MP	document creation
0.2	9.25.2018	VT	Copy and paste from ADS
0.3	1.12.2016	AT, GH	release candidate 1
1.0	1.20.2016	AT, GH, CB	official release
1.1	1.31.2016	AL	added design review requests

Contents

List of Figures

List of Tables

1 Introduction

Your introduction should provide a brief overview of the product concept and a reference to the requirement specification and architectural design documents in 1 or 2 paragraphs. The purpose is to provide the reader with the location of relevant background material that lead to the design details presented in this document.

2 System Overview

This section should reintroduce the full data flow diagram from the architectural specification, and discuss at a high level the purpose of each layer. You do not need to include a subsection for each layer, a 1 - 2 paragraph recap is sufficient.

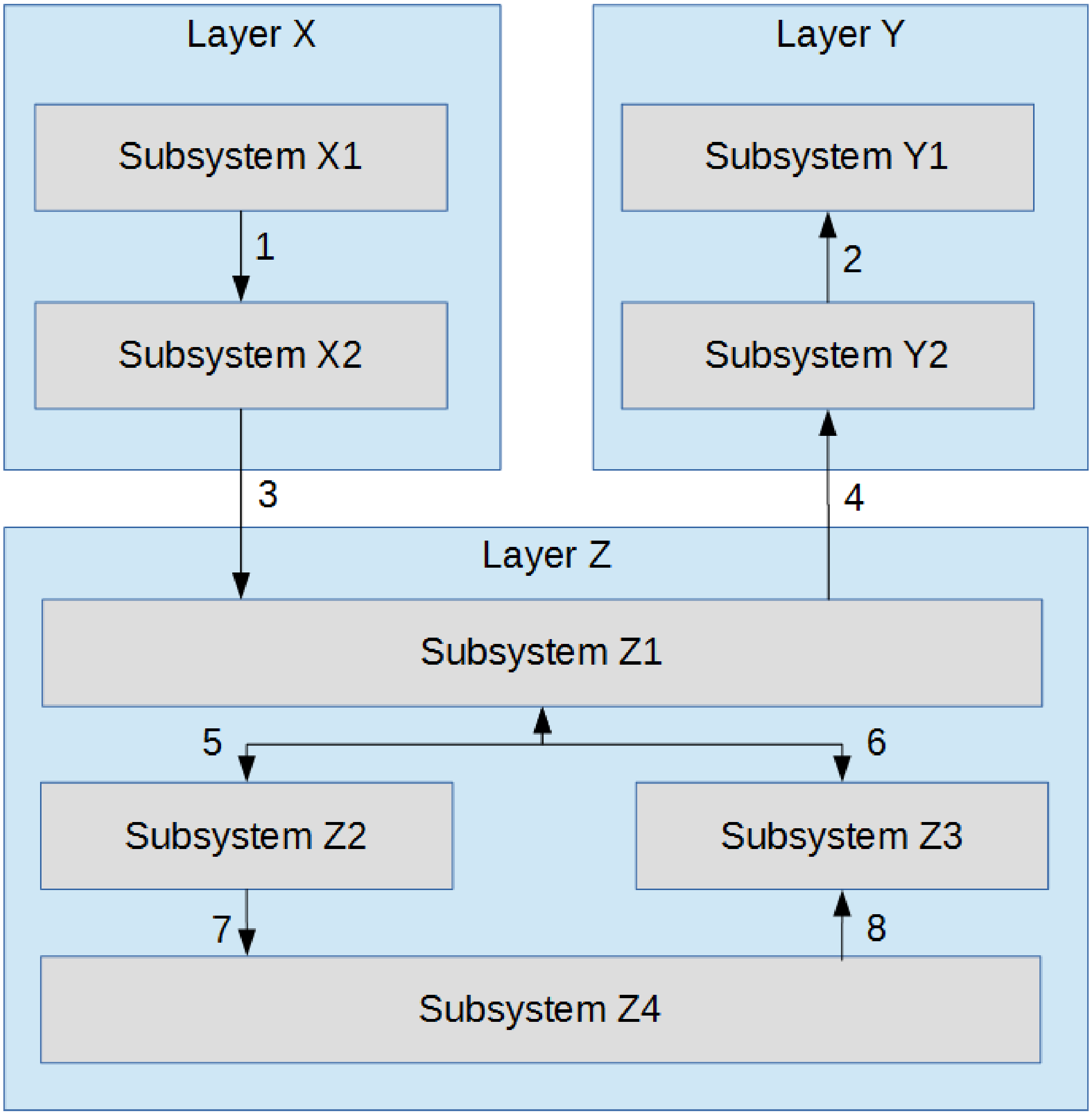

Figure 1: System architecture

3 Subsystem Definitions & Data Flow

This section breaks down your layer abstraction to another level of detail. Here you grapically represent the logical subsytems that compose each layer and show the interactions/interfaces between those subsystems. A subsystem can be thought of as a programming unit that implements one of the major functions of the layer. It, therefore, has data elements that serve as source/sinks for other subsystems. The logical data elements that flow between subsystems need to be explicitly defined at this point, beginning with a data flow-like diagram based on the block diagram.

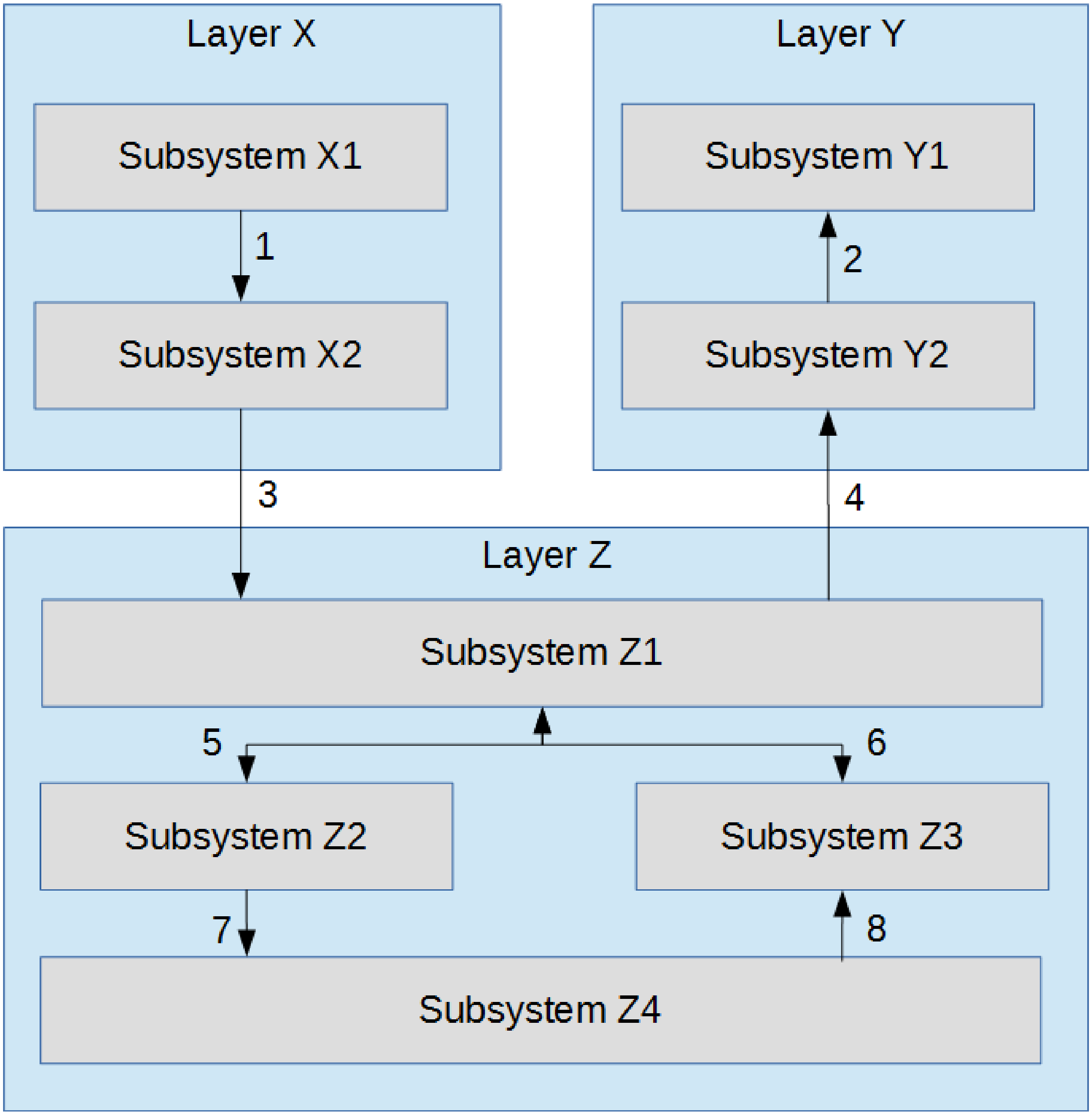

Figure 2: A simple data flow diagram

4 Brain Layer Subsystems

The Brain layer serves as a central hub for data exchange in the device. It should be controlling when hops are added to the system and for how long they steep. It will also be responsible for taking commands from the user.

4.1 Raspberry Pi

This serves as the main control device for the unit. It will host the webpage and run all calculations for the device. It will also activate the motors.

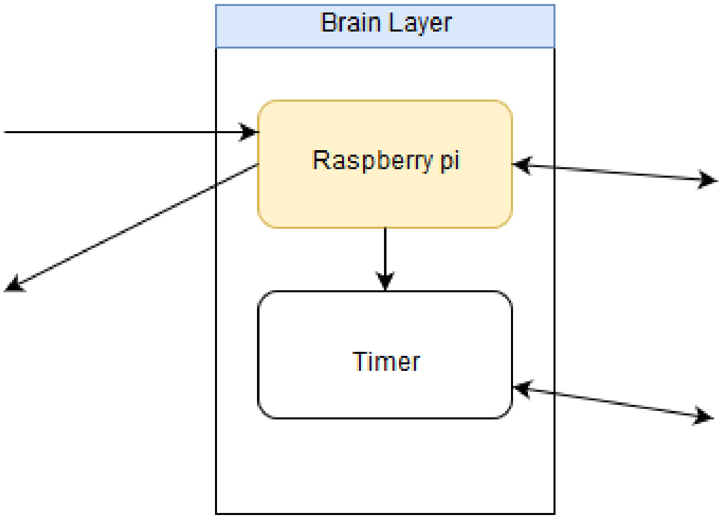

Figure 3: Raspberry Pi Subsystem

4.1.1 Assumptions

The assumption we are making is that the raspberry pi can securely and reliably host webpages and make calculations quickly enough for this system.

4.1.2 Responsibilities

The Raspeberry pi will be responsible for all server side calculations as well as the main functionality of the device out side of the user interface.

4.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#00	Controls the timer	Timer Control	output 1

4.2 Timer

This subsystem is soly responsible for keeping the time. Since this is an incredibly important function for brewing we must make sure we do it correctly.

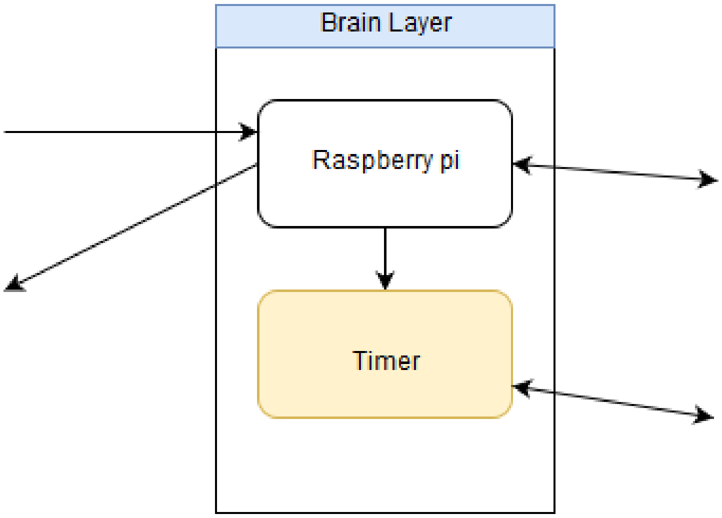

Figure 4: Timer

4.2.1 Assumptions

We assume it is possible to keep accurate time.

4.2.2 Responsibilities

This is responsible for the timed events that must happen at certain times.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Controls the timer	Signal Reception	input 1

5 Misc Layer Subsystems

These are analog devices that have no input or output yet matter for a fully functioning system.

5.1 Kettle

The kettle which holds the beer as it brews.

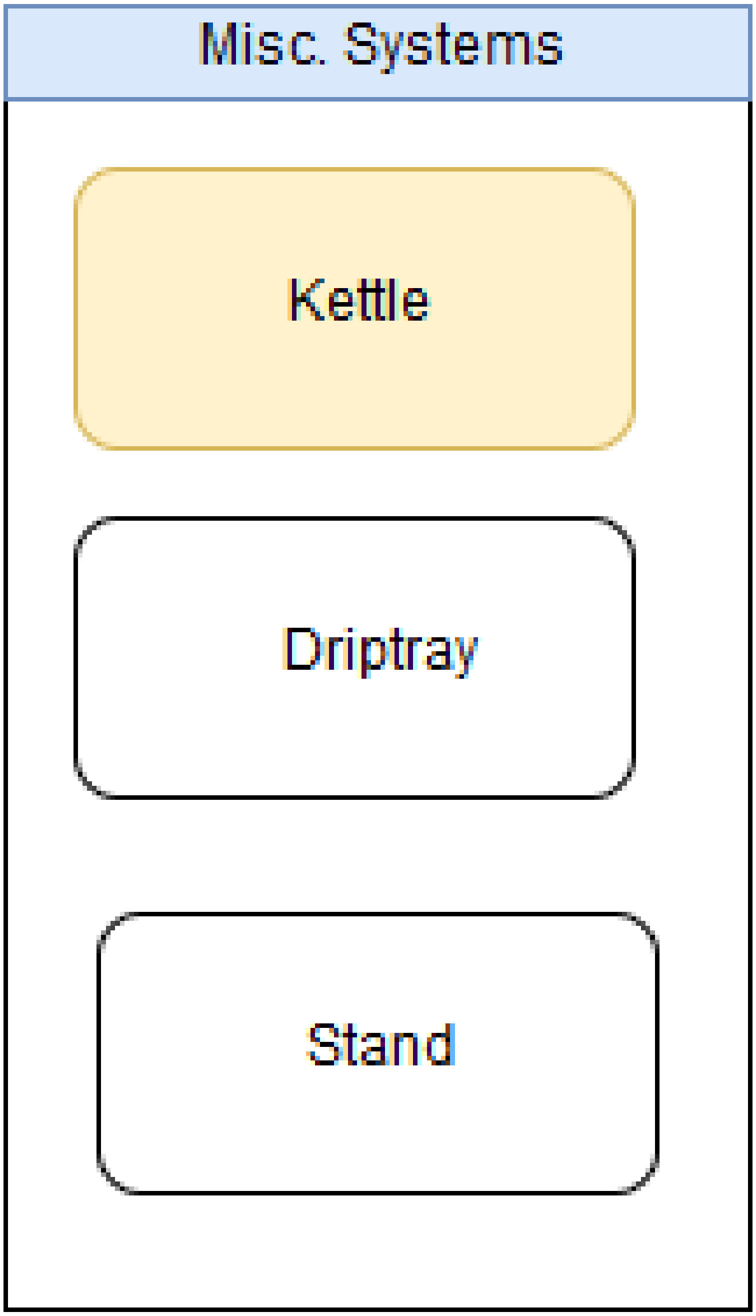

Figure 5: Example subsystem description diagram

5.1.1 Assumptions

None

5.1.2 Responsibilities

None

5.1.3 Subsystem Interfaces

None

5.2 Drip Tray

A machine washable tray which protects the system from dripping liquids post brewing.

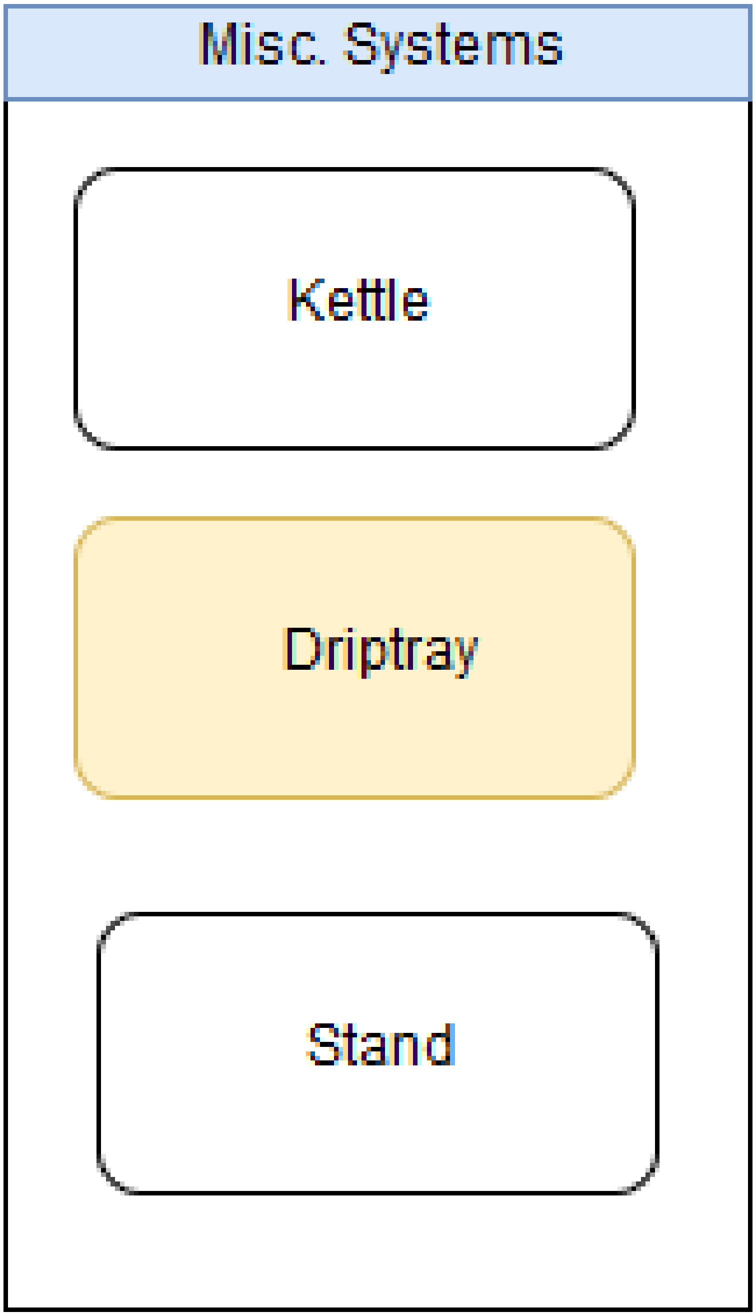

Figure 6: Example subsystem description diagram

5.2.1 Assumptions

None

5.2.2 Responsibilities

None

5.2.3 Subsystem Interfaces

None

5.3 Stand

housing which allows the device to stand tall enough too fulfill all it's purposes.

5.3.1 Assumptions

None

5.3.2 Responsibilities

None

5.3.3 Subsystem Interfaces

None

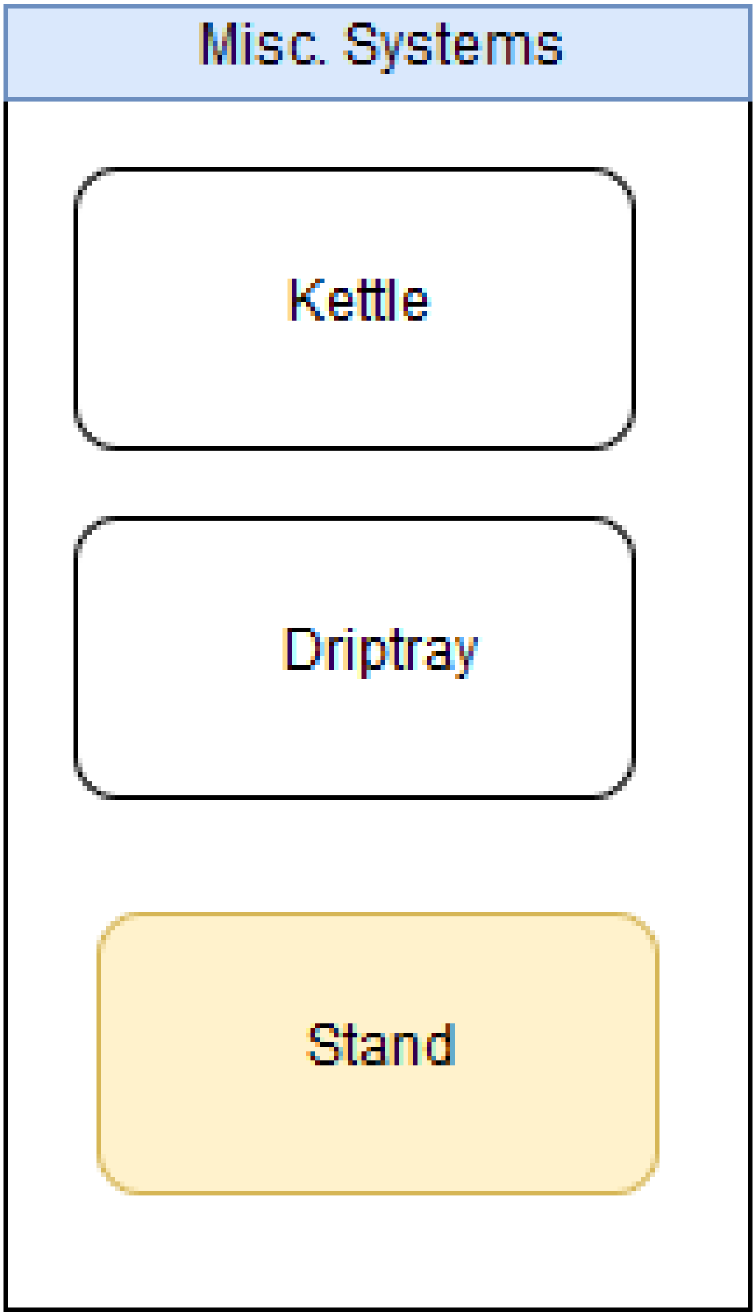

Figure 7: Example subsystem description diagram

6 Motor Control Layer Subsystems

The Motor Control layer is the layer where the physical action happens. This layer will have two subsystems, the first system is Motors. The Motors sub-system will have approximately ten motors that will control, what The Brew Crew likes to call the DOOM Hopper, using a Raspberry Pi. The DOOM Hopper will be in charge of dropping the brew hops into the kettle at a specific time. The second subsystem is Error Detection. The Error Detection subsystem will be keeping track of all the motors. In other words, the subsystem will check if all the motors are moving at the programmed rate and be positioned at the correct position.

6.1 Error Detection

The error detection subsystem will be powered the Brain layer, specifically, by the raspberry pi. Error detection will always be checking on the motors of the system, it will check the RPMs of the motors and fix the RPM if too slow or too fast. Error detection will also keep track of the position of the motor; with proper position of the motors the hops can be dropped correctly into the boiling kettle.

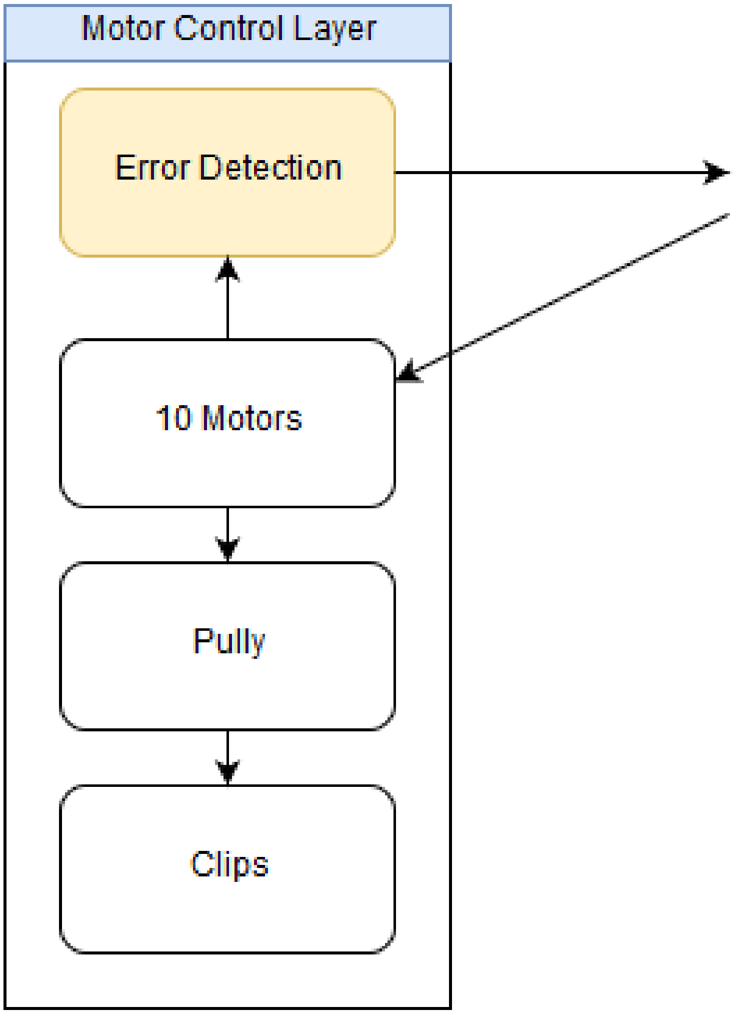

Figure 8: Error Detection Diagram

6.1.1 Assumptions

We assume error detection will be controlled by the Raspberry Pi.

6.1.2 Responsibilities

Error detection is responsible for keeping all motors at the right RPM and in the right position.

6.1.3 Subsystem Interfaces

The subsystem interface is the raspberry pi from the Brain Layer.

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Detects errors	Raspberry Pi	Sends positive or negative confirmation to Raspberry PI

6.2 Motors

The motors are the moving forces of the hop system. The objective of the motors is to move the hop dispensers into positon for displacing the hops and specialty grains into the stainless steel kettle for boiling. The user will select a recipe from the interface layer, which consist of a web page or an android application. Upon selection, only the hops in the dispensers needed for the recipe will be dumped into the kettle. This is important for the motor control since the user selection will determine how long the motors should run in order to position the right hop dispensers in front of the kettle.

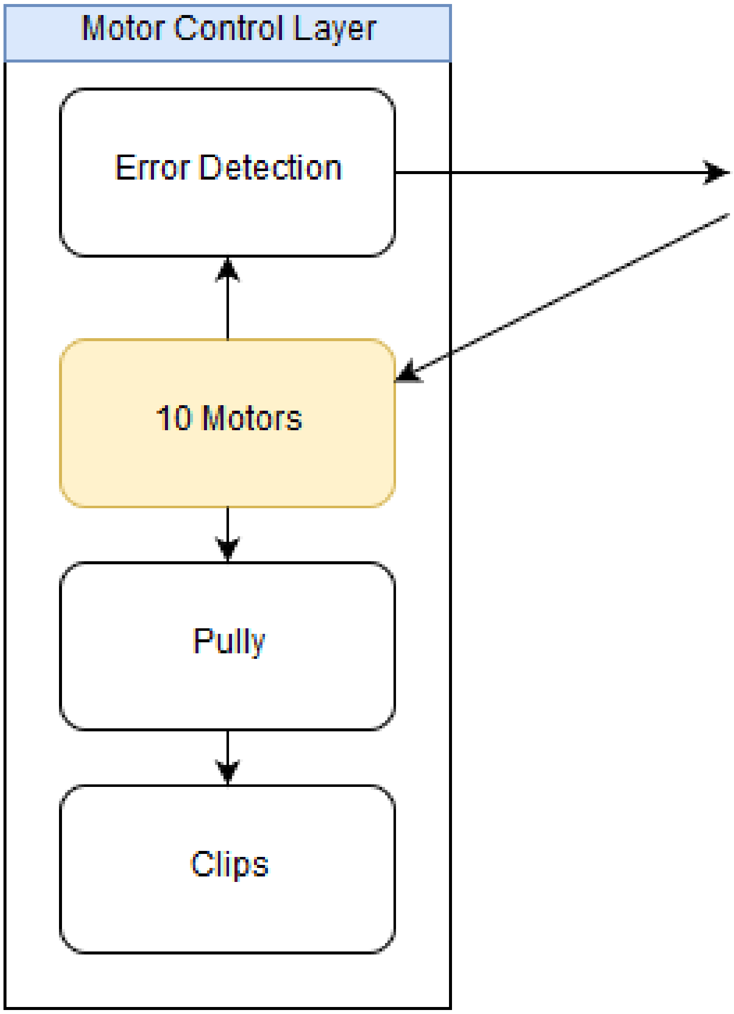

Figure 9: Motor Control Diagram

6.2.1 Assumptions

The assumptions for the motor subsystem is that there will be approximately 10 motors controlling the hop containers. Also, the motors will be controlled by the brain system, which consist of a raspberry pi.

6.2.2 Responsibilities

The motor subsystem is responsible for turning each hop container at a certain rpm. It will allow for the hops to be dumped at the right position.

6.2.3 Subsystem Interfaces

The subsystem's interface is the raspberry pi and the timer, which come from the Brain Layer.

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Motor Control	Raspberry Pi	Hop Dispenser Position

7 Interface Subsytem

The interface layer will serve as the software structure of the brewing system. It will act similar to a menu for the user to choose their recipe through the android app and will connect to a server from the raspberry pi or a webpage, possibly both.

7.1 Android App

The Android App will communicate with the raspberry pi to basically provide an interface for the user to utilize the hop machine.

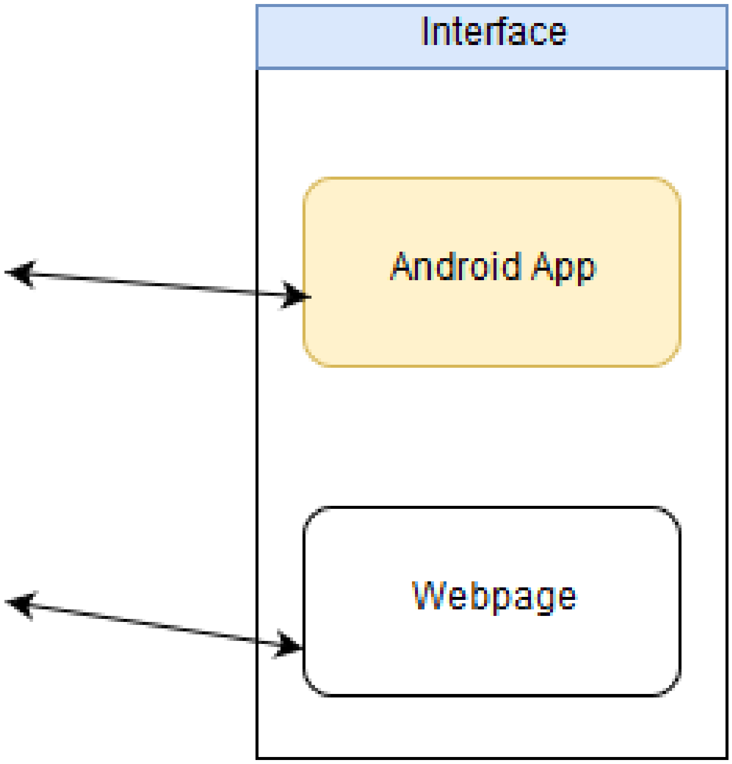

Figure 10: Example subsystem description diagram

7.1.1 Assumptions

The android app will provide recipes to choose from and communicate with the brain.

7.1.2 Responsibilities

The android app should have a menu like interface for the user to be able to easily understand and use. The app should be in sync with the brain at all times.

7.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Start and Stop	User Input	Signal to Pi
#02	Send recipe to the webpage	User Input File	Parsed recipe class
$03	Receive updates from webpage	Status class	Updated status class

7.2 Webpage

It will act as the same thing as the android app and as a possible substitute. It will allow users to access the app and alternately anything connected to the internet.

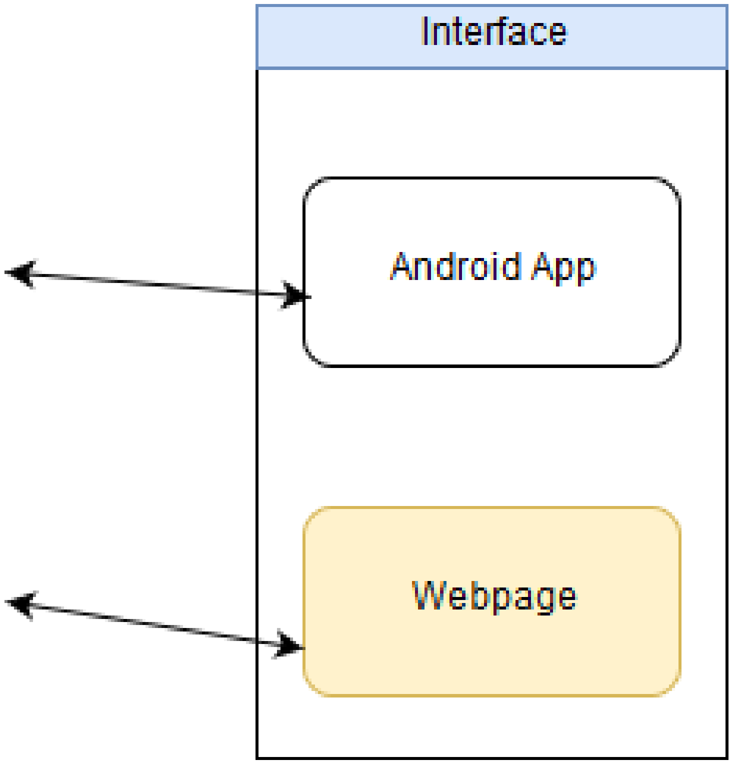

Figure 11: Example subsystem description diagram

7.2.1 Assumptions

It basically does the same thing as the android app and will allow users to use either app other devices to access the brewing system. Will give the user more direct control over the entire system.

7.2.2 Responsibilities

The webpage will be a single page ran on the Raspberry Pi. This subsystem will be able to display any important information and manually control the system. The page will be split up into sections to help the user understand the system in its current state.

7.2.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Send signal to dispense hops	User input	Send a command
#02	Display updates	Data class from Pi	N/A
#03	Send new recipes to database	Recipe File	Parsed File